IST 210 Organization of Data

Database and the Web
References

- ASP Tutorial from MSDN
HTML

VB Script

SQL
Create Dynamic Web Applications

- Static Web application
  - Request with a URL (e.g., http://www.psu.edu)
    Which contains three components: protocol, web server name, and folder path to an HTML page
  - Server simply send back the page

- From static to dynamic web pages
  - Take user input and respond accordingly
  - Allow access to information stored in a database
    - https://aspdb.aset.psu.edu/ist210tsb4/example.asp
    - https://aspdb.aset.psu.edu/ist210tsb4/student.html
    - https://aspdb.aset.psu.edu/ist210tsb4/studentlist.asp
Web Pages with Database Contents

- Web pages contain the results of database queries. How do we generate such pages?
  - Common Gateway Interface (CGI)
    - Web server creates a new process when a program interacts with the database.
    - Web server communicates with this program via CGI (Common gateway interface)
    - Program generates result page with content from the database

**Problem**: need to run multiple processes which is not efficient.
Application Servers

- In CGI, each page request results in the creation of a new process → generally inefficient
- Application server: Piece of software between the web server and the applications
- Functionality:
  - Hold a set of threads or processes for performance
  - Database connection pooling (reuse a set of existing connections)
  - Integration of heterogeneous data sources
  - Transaction management involving several data sources
  - Session management
Other Server-Side Processing

- Java Servlets: Java programs that run on the server and interact with the server through a well-defined API.
- JavaBeans: Reusable software components written in Java.
- Java Server Pages and Active Server Pages: Code inside a web page that is interpreted by the web server
Active Server Pages (ASP)

- ASP is programming model that allows dynamic, interactive Web pages to be created on server.
- ASP runs in-process with the server, and is optimized to handle large volume of users.
- When an `.asp` file is requested, Web server calls ASP, which reads requested file, executes any commands, and sends generated HTML page back to browser.
Active Server Pages (ASP)
ASP Code

- Combination of three types of syntax:
  - Text
  - HTML tags
  - ASP scripts
ASP Scripts

- ASP scripts can be written in
  - VBScript
    
    <SCRIPT LANGUAGE=VBScript>
  - JavaScript
    
    <SCRIPT LANGUAGE=JavaScript>
  - ActiveX Components

- Client-side vs. Server-Side
  - Client-side scripts downloaded to and execute on the client machine. (Problems: features by not be supported by some browsers)
  - Server-side scripts
    
    Run directly on the server and generate data to be viewed by the browser in HTML. No concern for browser capability.
ASP Code

- Script codes are executed by the server
- Generate HTML, on-the-fly, when requested
- ASP code is browser independent.
- ASP code can be viewed at the server using Text Editor
- Browser can not directly view the source code of a ASP program
ActiveX Data Objects (ADO)

- Programming extension of ASP supported by Microsoft IIS for database connectivity.
- Supports following key features:
  - Independently-created objects.
  - Support for stored procedures.
  - Support for different cursor types.
  - Batch updating.
  - Support for limits on number of returned rows.
- Designed as an easy-to-use interface to OLE DB.
Getting User Input From a Form

- **Connection** – establishing link between application program and database
- **Recordset** – contains data returned from a specific action on the database
- **Command** – allow you to run commands against a database
Extensible Markup Language (XML)
Question:

What’s the difference between the world of documents and databases?
## Documents vs Databases

### Document world
- plenty of small documents
- usually static
- implicit structure
  - section, paragraph
- tagging
- human friendly
- content
  - form/layout, annotation
- Paradigms
  - “Save as”, wysiwyg
- meta-data
  - author name, date, subject

### Database world
- a few large databases
- usually dynamic
- explicit structure (schema)
- records
- machine friendly
- content
  - schema, data, methods
- Paradigms
  - Atomicity, Concurrency, Isolation, Durability
- meta-data
  - schema description
What to do with them

Documents
- editing
- printing
- spell-checking
- counting words
- retrieving
- searching

Database
- updating
- cleaning
- querying
The thin line

- The line between the document world and the database world is not clear.
- In some cases, both approaches are legitimate.
- An interesting middle ground is data formats -- of which XML is an example
A common form of data extraction

Find the names and telephones of all employees in IST
Lineage (WWW Consortium)

Ease of Use

Flexibility

Standard Generalized Markup Language (SGML – Late 1980s)

Extensible Markup Language (XML – Late 1990s)

Hypertext Markup Language (HTML – Early 1990s)
Need

- Doctor want to who wants to send you medical record to a specialist:

  <html>
  <p>Patient G. Washington is allergic to penicillin</p>
  </html>

- As HTML provides a way for all computers to read Internet documents, but how can a computer read the data?
HTML

- Lingua franca for publishing hypertext on the World Wide Web
- Designed to describe how a Web browser should arrange text, images and push-buttons on a page.
- Easy to learn, but does not convey structure.
- Fixed tag set.

```
<HTML>
  <HEAD><TITLE>Welcome to IST210</TITLE></HEAD>
  <BODY>
    <H1>Introduction</H1>
    <IMG SRC="ist.jpeg" WIDTH="200" HEIGHT="150" />
  </BODY>
</HTML>
```
The Structure of XML

- XML consists of *tags* and *text*

- Tags come in pairs  `<date> ...</date>`

- They must be properly nested
  `<date> <day> ... </day> ... </date> --- good`
  `<date> <day> ... </date>... </day> --- bad`

(You can’t do `<i> ... <b> ... </i> ...</b> in HTML)
XML text

XML has only one “basic” type -- text.

It is bounded by tags e.g.

```
<title> G. Washington </title>
<year> 2001 </year> --- 2001 is still text
```

XML text is called PCDATA (for parsed character data). It uses a 16-bit encoding.

Later we shall see how new types are specified by XML-data
XML structure

Nesting tags can be used to express various structures. E.g. A tuple (record):

```xml
<person>
  <name> G. Washington </name>
  <tel> (703) 111 1000 </tel>
  <email> gw@mtvernon.com </email>
</person>
```
We can represent a list by using the *same* tag repeatedly:

```
<addresses>
  <person> ... </person>
  <person> ... </person>
  <person> ... </person>
  ...
</addresses>
```
The segment of an XML document between an opening and a corresponding closing tag is called an *element*.

```
<person>
  <name> G Washington </name>
  <tel> (703) 111 1000 </tel>
  {<tel> (703) 111 1001 </tel>}
  <email> gw@mtvernon.com </email>
</person>
```
XML is tree-like

- person
  - name: G Washington
  - tel: (703) 111 1000
  - tel: (703) 111 1001
  - email: gw@mtvernon.com
Mixed Content

An element may contain a mixture of sub-elements and PCDATA

<airline>
  <name>Agony Airways</name>
  <motto>
    US’s <dubious>favorite</dubious> airline
  </motto>
</airline>

Data of this form is not typically generated from databases. It is needed for consistency with HTML.
A Complete XML Document

<?xml version="1.0"?>

<person>
  <name> G Washington </name>
  <tel> (703) 111 1000 </tel>
  <email> gw@mtvernon.com </email>
</person>
Document Type Descriptors

Imposing structure on XML documents
Document Type Descriptors (DTDs) impose structure on an XML document.

There is *some* relationship between a DTD and a schema.

The DTD is a *syntactic* specification.
Example: The Address Book

\[
\langle \text{person} \rangle \\
\langle \text{name} \rangle \text{ MacNiel, John } \langle /\text{name} \rangle \\
\langle \text{greet} \rangle \text{ Dr. John MacNiel } \langle /\text{greet} \rangle \\
\langle \text{addr} \rangle \text{ 1234 Huron Street } \langle /\text{addr} \rangle \\
\langle \text{addr} \rangle \text{ Rome, OH 98765 } \langle /\text{addr} \rangle \\
\langle \text{tel} \rangle (321) \text{ 786 2543 } \langle /\text{tel} \rangle \\
\langle \text{fax} \rangle (321) \text{ 786 2543 } \langle /\text{fax} \rangle \\
\langle \text{tel} \rangle (321) \text{ 786 2543 } \langle /\text{tel} \rangle \\
\langle \text{email} \rangle \text{ jm@abc.com } \langle /\text{email} \rangle \\
\langle /\text{person} \rangle
\]

- Exactly one name
- At most one greeting
- As many address lines as needed (in order)
- Mixed telephones and faxes
- As many as needed
Specifying the structure

- **name** to specify a name element
- **greet?** to specify an optional (0 or 1) greet elements
- **name,greet?** to specify a name followed by an optional greet
Specifying the structure (cont)

- **addr*** to specify 0 or more address lines
- **tel | fax** a tel *or* a fax element
- **(tel | fax)*** 0 or more repeats of tel or fax
- **email*** 0 or more email elements
Specifying the structure (cont)

So the whole structure of a person entry is specified by

name, greet?, addr*, (tel | fax)*, email*

This is known as a regular expression.
Summary

- XML is a new data format. Its main virtues are:
  - widespread acceptance and the ability to handle semistructured data (data without schema)
- The emerging combination of database and XML provide a powerful tool for delivering content over the web